Amendments to the Specification:

Please replace the paragraph on page 2 lines 8 to 19 with the following amended paragraph:

Network data storage is most economically provided by an array of low-cost disk drives integrated with a large semiconductor cache memory. A number of data mover computers are used to interface the cached disk array to the network. The data mover computers perform file locking and file metadata management and mapping of the network files to logical block addresses of storage in the cached disk array, and move data between network clients and storage in the cached disk array. Typically the logical block addresses of storage are subdivided into logical volumes. Each logical volume is mapped to the physical storage using a respective striping and redundancy scheme. The data mover computers typically use the Network File System (NFS) protocol to receive file access commands from ~~UNIX and Linux~~ clients using the UNIX (Trademark) operating system or the LINUX (Trademark) operating system, and the Common Internet File System (CIFS) protocol to receive file access commands from ~~MicroSoft (MS) Windows~~ clients using the MicroSoft (MS) WINDOWS (Trademark) operating system.

Please replace the paragraph on page 2 line 20 to page 3 line 8 with the following amended paragraph:

More recently there has been a dramatic increase in various ways of networking clients to storage and protocols for client access to storage. These networking options include a Storage Area Network (SAN) providing a dedicated network for clients to access storage devices directly via Fibre-Channel, and Network Attached Storage (NAS) for clients to access storage over a Transmission Control Protocol (TCP) and Internet Protocol (IP) based network. In addition to the high-level file-access protocols such as NFS and CIFS, the various networking options may use lower-level protocols such as the Small Computer System Interface (SCSI), the Fibre-Channel protocol, and SCSI over IP ~~(iSCSI)~~. However, most network facilities for data sharing and protection are based on file access protocols, and therefore the use of lower-level protocols in lieu of file access protocols for access to network storage may limit the available options for data sharing and protection.

Please replace the paragraph on page 6 line 2 with the following amended paragraph:

FIG. 11 is a table of server opcodes for the NBS protocol of FIG. [[11]] 9;

Please replace the paragraph on page 7 line 20 to page 8 line 6 with the following amended paragraph:

As introduced above, some clients may desire to use lower-level protocols such as the Small Computer System Interface (SCSI), the Fibre-Channel protocol, and the SCSI over IP ~~(iSCSI)~~ protocol in order to access network storage. One environment where this is desirable is

a Microsoft Exchange platform. In this environment, a Microsoft Exchange server, or a server for a database such as an Oracle or SQL database, typically stores its database component files and tables such as storage groups, and transaction logs to one or more block devices. It is desired to replace these block devices with remote block devices in a network file server, and to provide disaster protection by replicating the database files and transaction logs to a geographically remote network file server and taking read-only copies or snapshots of the database and logs, for backup to tape.

Please replace the paragraph on page 8 lines 17-16 with the following amended paragraph:

For the data processing network in FIG. 2, for example, the client may use ~~iSCSI~~ the SCSI over IP protocol over the IP network 20. In this example, the software modules in the client 23 include application programs 51 layered over an operating system 52. The operating system manages one or more file systems 53. To access the network storage, the file system routines invoke a SCSI device driver 54, which issues SCSI commands to an ~~iSCSI~~ SCSI over IP initiator 55. The ~~iSCSI~~ SCSI over IP initiator inserts the SCSI commands into a TCP connection established by a TCP/IP module 56. The TCP/IP module 56 establishes the TCP connection with the data mover 26, and packages the SCSI commands in IP data packets. A network interface card 57 transmits the IP data packets over the IP network 20 to the data mover 26.

Please replace the paragraph on page 8 line 17 to page 9 line 3 with the following amended paragraph:

A network interface card 61 in the data mover 26 receives the IP data packets from the IP network 20. A TCP/IP module 62 decodes data from the IP data packets for the TCP connection and sends it to an ~~iSCSI~~ SCSI over IP target software driver module 63. The ~~iSCSI~~ SCSI over IP target module 63 decodes the SCSI commands from the data, and sends the SCSI commands to a SCSI termination 64. The SCSI termination is a software module that functions much like a controller in a SCSI disk drive, but it interprets a storage object 65 that defines a logical disk drive. The SCSI termination presents one or more virtual LUNs to the ~~iSCSI~~ SCSI over IP target 63. A virtual LUN is built on top of the storage object 65, and it emulates a physical SCSI device by implementing SCSI primary commands (SPC-3) and SCSI block commands (SBC-2).

Please replace the paragraph on page 9 lines 4-11 with the following amended paragraph:

Instead of reading or writing data directly to a physical disk drive, the SCSI termination 64 reads or writes to a data storage area of the storage object 65. The storage object, for example, is contained in a file or file system compatible with the UNIX operating system and the MS-Windows operating system. Therefore, file access protocols such as NFS and CIFS may access the storage object container file. Consequently, conventional facilities for data sharing and protection may operate upon the storage object container file. Use of a file as a container for

the storage object may also exploit some file system features such as quotas, file system cache in the data mover, and block allocation on demand.

Please replace the paragraph on page 9 lines 12-16 with the following amended paragraph:

The ~~iSCSI~~ SCSI over IP protocol begins with a login process during which the ~~iSCSI~~ SCSI over IP initiator establishes a session with a target. TCP connections may be added and removed from a session. The login process may include authentication of the initiator and the target. The TCP connections are used for sending control messages, and SCSI commands, parameters, and data.

Please replace the paragraph on page 9 line 17 to page 10 line 10 with the following amended paragraph:

FIG. 3 shows one type of an ~~iSCSI~~ SCSI over IP protocol PDU command 82. The command 82 includes a one-byte opcode indicating the command type, and two bytes of flags. The first byte of flags includes two flags that indicate how to interpret the following length field, and a flag set to indicate a read command. The second byte of flags includes one Autosense flag and three task attribute flags. The command 82 includes a length indicating the length of the command in bytes, and a Logical Unit Number (LUN) specifying the Logical Unit to which the command is targeted. The command 82 includes an Initiator Task Tag assigned to each SCSI

task initiated by the SCSI initiator. A SCSI task is a linked set of SCSI commands. The Initiator Task Tag uniquely identifies each SCSI task initiated by the SCSI initiator. The command 82 includes a Command Reference Number (CMDRN) for sequencing the command, and an Expected Status Reference Number (EXPSTATRN) for indicating that responses up to EXPSTATRN-1 (mod 2**32) have been received. The command 82 includes an Expected Data Transfer Length that the SCSI initiator expects will be sent for this SCSI operation in SCSI data packets. The command 82 includes a 16-byte field 83 for a Command Descriptor Block (CDB). The command 82 may also include additional command-dependent data.

Please replace the paragraph on page 11 line 9-18 with the following amended paragraph:

The storage object attributes 86 also include a storage capacity in bytes, and the amount of storage presently used, and the amount of free space in the storage object. The storage object attributes 86 include a list of users permitted to access the storage object through the SCSI termination module (64 in FIG. 2), and a respective permission and quota for each user. Moreover, the storage object attributes may include configuration information, such as a location (bus, target and LUN) of the storage object, and an internal organization of the storage object, such as a level of redundancy in an array of disk drives (RAID level) and a striping scheme. The specified internal organization of the storage object could be used as a guide or specification for mapping of the data storage area 87 of the container file ~~87~~ 84 to storage in the cached disk array (49 in FIG. 2).

Please replace the paragraph on page 13 line 18 to page 14 line 3 with the following amended paragraph:

In the data processing system of FIG. 2, it is desired to permit the client 23 to manage backup and replication of its SCSI storage object in the data mover 26 during concurrent access to the storage object using the ~~iSCSI~~ SCSI over IP protocol. For example, while the client 23 writes data to the data mover 26, the data mover 26 replicates the data to the second network file server 22 in FIG. 1 by transmitting a copy of the data over the IP network 20 using the NFS or CIFS protocols. One way of doing this is to provide a parallel and concurrent TCP connection between the client 23 and the data mover 26 for control of snapshot copy and IP replication applications in the data mover 26. This method is described below with reference to FIGS. 7 to 14.

Please replace the paragraph on page 14 lines 13-18 with the following amended paragraph:

One way of pausing write access to the storage object 65 at the completion of a commit operation is to provide a service in the applications 51 or the file system 53 that provides a notification to interested applications of the commit operation and suspends further write operations to storage until an acknowledgement is received from the interested applications. Although the Windows operating system ~~53~~ 52 does not presently provide such a service, the Microsoft Exchange application provides such a service.

Please replace the paragraph on page 14 line 19 to page 15 line 18 with the following amended paragraph:

In a MS Windows machine, the Windows Management Instrumentation (WMI) facility 73 provides a mechanism for communication between processes. The WMI facility 73 functions as a mailbox between processes in the client 23. A process may call a WMI driver routine that places data into the WMI facility and notifies subscribers to the data. In the example of FIG. 7, for example, the virtual block device manager 71 calls a routine in a snapshot and replication dynamic link library (DLL) 72, which receives notification of a commit event. For example, the Microsoft Exchange application responds to an application program interface (API) call that invokes the service in Exchange to suspend further write operations after a commit operation, and returns a notification that further write operations have been suspended. A similar API is used in UNIX file systems. This API call is provided in order to put the database such as Exchange or Oracle in a quiescent state in order to make a backup copy of the database. In the event of a system crash, the database application can replay its logs during recovery to ensure that its backup database is brought back to a consistent state. When a commit event has occurred and further writing over the ~~iSCSI/TCP~~ SCSI over IP TCP connection (112 in FIG. 12) is inhibited, a network block services (NBS) driver 74 in the client establishes a parallel and concurrent TCP connection (113 in FIG. 12) to a network block services server 75 in the data mover (~~21~~ 26 in ~~FIGS. 11 and~~ FIG. 12.) NBS control commands cause a snapshot copy facility 76 or an IP replication facility 77 to initiate a snapshot copy or IP replication process upon the

storage object 65. The snapshot copy or IP replication process may continue as a background process concurrent with subsequent write access on a priority basis when the SCSI termination 64 executes SCSI write commands from the client's SCSI driver 54.

Please replace the paragraph on page 16 lines 3-20 with the following amended paragraph:

The NBS protocol is introduced in Xiaoye Jiang et al., "Network Block Services for Client Access of Network-Attached Data Storage in an IP Network," U.S. Patent Application Ser. 10/255,148 filed Sep. 25, 2002, incorporated herein by reference. This protocol is extended for snapshot copy and replication of storage objects, as further described below with reference to FIGS. 9 to 11. Details of a snapshot copy facility are described in Keedem U.S. Patent 6,076,148 issued June 13, 2000, incorporated herein by reference; and Philippe Armangau et al., "Data Storage System Having Meta Bit Maps for Indicating Whether Data Blocks are Invalid in Snapshot Copies," U.S. Patent Application Ser. 10/213,241 filed Aug. 6, 2002, incorporated herein by reference. Details of an IP replication facility are described in Raman, et al., U.S. Patent Application Ser. No. 10/147,751 filed May 16, 2002, entitled "Replication of Remote Copy Data for Internet Protocol (IP) transmission," incorporated herein by reference; and Philippe Armangau et al., Data Recovery With Internet Protocol Replication With or Without Full Resync, U.S. Patent Application Ser ~~No. _____ filed~~ No. 10/603,951 filed June 25, 2003, incorporated herein by reference. The snapshot copy or IP replication facility, for example, operates on a file system compatible with the UNIX and MS Windows operating

systems. In this case, the snapshot copy facility 76 or the IP replication facility 77 accesses the storage object container file 84 through the UxFS file system 44 in the data mover 26.

Please replace the paragraph on page 17 line 10 to page 18 line 11 with the following amended paragraph:

The network block services driver 74 communicates with the network block services server 75 using a relatively light-weight protocol designed to provide block level remote access of network storage over TCP/IP. This protocol also provides remote control of snapshot copy and IP replication facilities. The network block services server 75 maintains in memory a doubly-linked list of storage objects accessible to clients via their network block services drivers. Each storage object is also linked to a list of any of its snapshot copies. A copy of this list structure is maintained in storage. When the data mover 26 reboots, the NBS server rebuilds the in-memory list structure from the on-disk structure. The data mover 26 also maintains a directory of the storage objects using as keys the file names of the storage object container files. The in-memory list structure and the directory are extended to include the ~~iSCSI~~ SCSI over IP storage objects, so that each ~~iSCSI~~ SCSI over IP storage object is accessible to a client through the SCSI termination 64 or the network block services server 75. In particular, each virtual LUN recognized by the SCSI termination 64 has a corresponding NBS identifier recognized by the network block services server 75 and a corresponding storage object container file name. API calls are provided to coordinate the ~~iSCSI~~ SCSI over IP initiator 66 and the SCSI termination 64 with the NBS protocol during snapshot operations. For example, the snapshot and replication

DLL 72 includes an API call through the WMI 73 to the ~~iSCSI~~ SCSI over IP initiator 66 for changing the destination address of the ~~iSCSI~~ SCSI over IP protocol. This API call can be used during a restore operation, in order to resume processing from a backup copy of the storage object 65 after a disruption. The storage object 65 could be included in a storage object container file or could be a raw volume of the storage array or any combination of volumes such as raw volumes, slices, striped volumes or meta concatenated volumes. This approach has minimal impact on upper layer components of the operating system of the client 23.

Please replace the paragraph on page 18 lines 11-14 with the following amended paragraph:

FIG. 9 shows an IP data packet encoded by the network block services driver (74 in FIG. [[6]] 7). The data packet includes a packet header [[80]] 100 and, when appropriate, data [[81]] 101 appended to the packet header. The packet header, for example, has the following format:

Please replace the paragraph on page 28 line 19 to page 29 line 13 with the following amended paragraph:

In a first step 121 of FIG. 13, the virtual block device manager receives a snapshot or replication request from the system administrator or another application program of the client. In step 122, the virtual block device manager invokes the DLL routine for a snapshot or replication

of the virtual block device. In step 123, the call of the routine in the Windows operating system, or a kernel call in the UNIX operating system, for a snapshot or replication of the virtual block device initiates a sync and suspend ~~iSCSI~~ SCSI over IP application interface (API) call to WMI 73. This call is relayed to the Exchange application (111 in FIG. 12). Similar calls would be relayed to other applications using virtual block devices to be snapshotted or replicated. Then in step 124 the virtual block device manager sets a timer and then suspends its execution, until execution is resumed by receiving a callback notification that Exchange or other applications have completed a sync and suspend operation, or by expiration of the timer. In step 125, if execution has been resumed but no callback was received, then an error is logged indicating that the Exchange application has failed to perform the sync and suspend ~~iSCSI~~ SCSI over IP operation within the timer interval. Otherwise, if a callback has been received, then execution continues to step 126. In step 126, the virtual block device manager sends a snapshot or replicate command to the data mover via the NBS TCP connection. After step 126, execution continues in step 127 of FIG. 14.

Please replace the paragraph on page 29 lines 14 to 21 with the following amended paragraph:

In step 127 of FIG. 14, the virtual block device manager sets a timer and suspends execution. Execution is resumed upon a callback from the network block services driver reporting that a snapshot or replication has been initiated, or upon expiration of the timer interval. In step 128, if execution has been resumed but no callback was received, then an error

is logged indicating that the data mover has failed to initiate a snapshot or replication within the timer interval. If a callback was received, then execution continues to step 129. In step 129, the DLL for snapshot or replication initiates resumption of the ~~iSCSI~~ SCSI over IP operation by the Exchange or other applications.

Please replace the paragraph on page 30 line 22 to page 31 line 5 with the following amended paragraph:

In view of the above, there has been described a method of containing a storage object such as a virtual disk drive or storage volume in a file in order to provide access to the storage object by a low-level protocol such as SCSI, ~~iSCSI~~ SCSI over IP, or FC concurrent with access to the container file by a high-level protocol such as NFS or CIFS. This permits block level access via different types of network connections such as SAN and NAS concurrent with file system sharing by clients with diverse operating systems, and fast file system backup, fail-over, and recovery.

Please replace the abstract on page 43 lines 2-12 with the following amended paragraph:

A storage object such as a virtual disk drive or a raw logical volume is contained in a UNIX compatible file so that the file containing the storage object can be exported using the NFS or CIFS protocol and shared among UNIX and MS Windows clients or servers. The storage object can be replicated and backed up using conventional file replication and backup facilities without disruption of client access to the storage object. For client access to data of the storage object, a software driver accesses the file containing the storage object. For example, a software driver called a virtual SCSI termination is used to access a file containing a virtual SCSI disk drive. Standard storage services use the ~~iSCSI~~ SCSI over IP protocol to access the virtual SCSI termination. An IP replication or snapshot copy facility may access the file containing the virtual SCSI disk drive using a higher-level protocol.